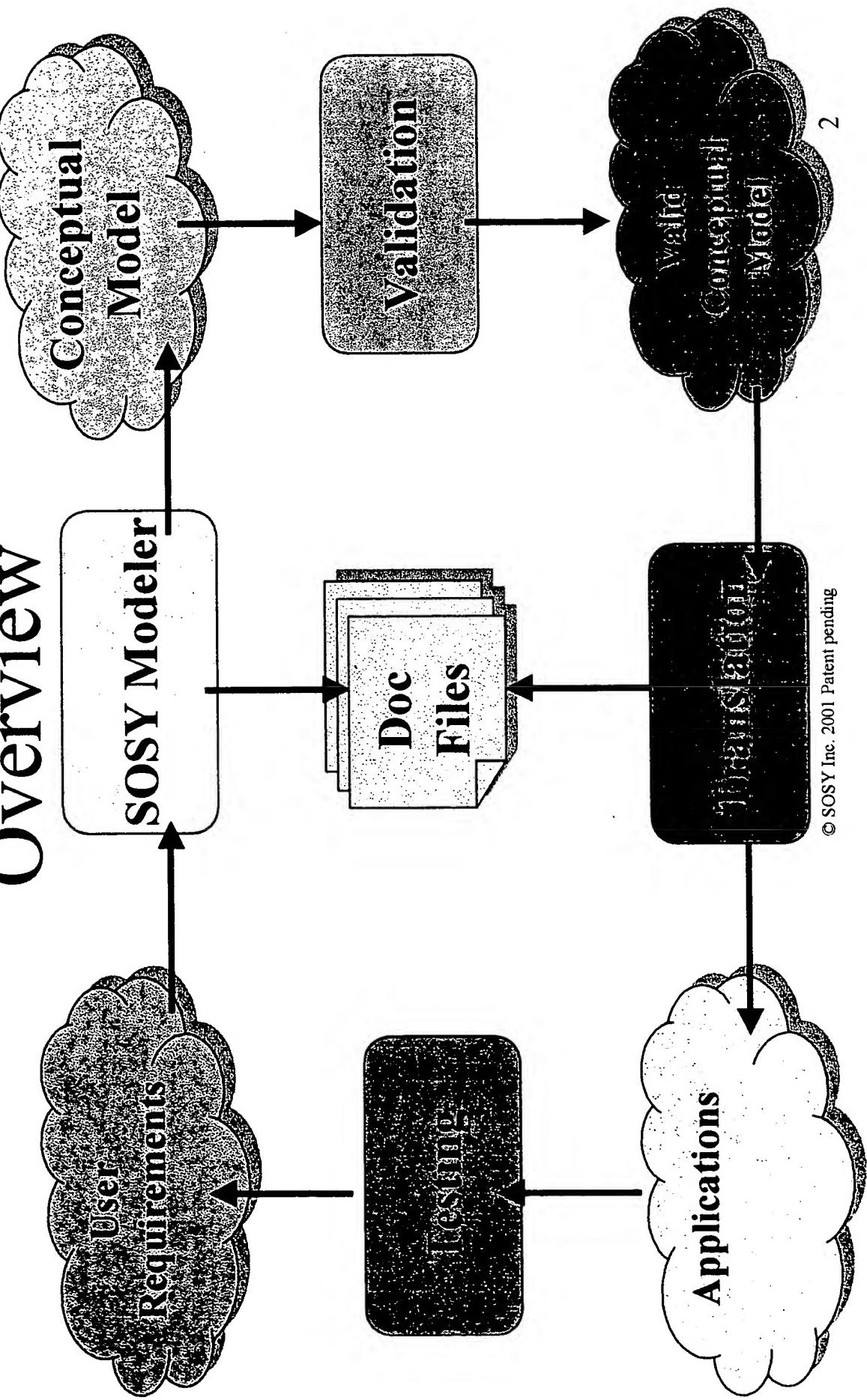


Summary

- Modeling
- Validation
- Documentation
- Persistence
- Business Logic
- User Interface

Overview



2

© SOSY Inc. 2001 Patent pending

Conceptual Modeling Phase

Index

- Intro
- Overview
- Phase 0. Requirements elicitation.
- Phase 1. Classes identification.
- Phase 2. Relationships between classes.
- Phase 3. Filling classes' details.

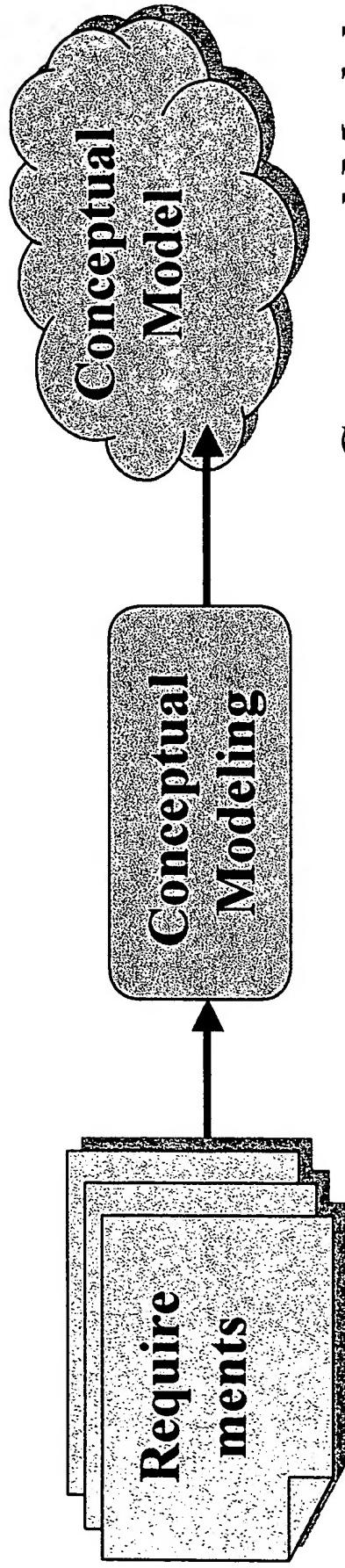
Index

- Phase 4. Express evaluations.
- Phase 5. Agent relationships.
- Phase 6. State Transition Diagram.
- Phase 7. Presentation Model.

Intro

- Conceptual Modeling Phase is a process of systematically & precisely description of the system to build, using:
 - Graphical UML compliant diagrams.
 - Constraints and semantics in a formal non-ambiguous language.
 - This phase is assisted by an integrated Modeler tool.

Overview



Requirements

- Specifications
- Documents
- Interviews
- Reports
- Other info. sources

Conceptual Model

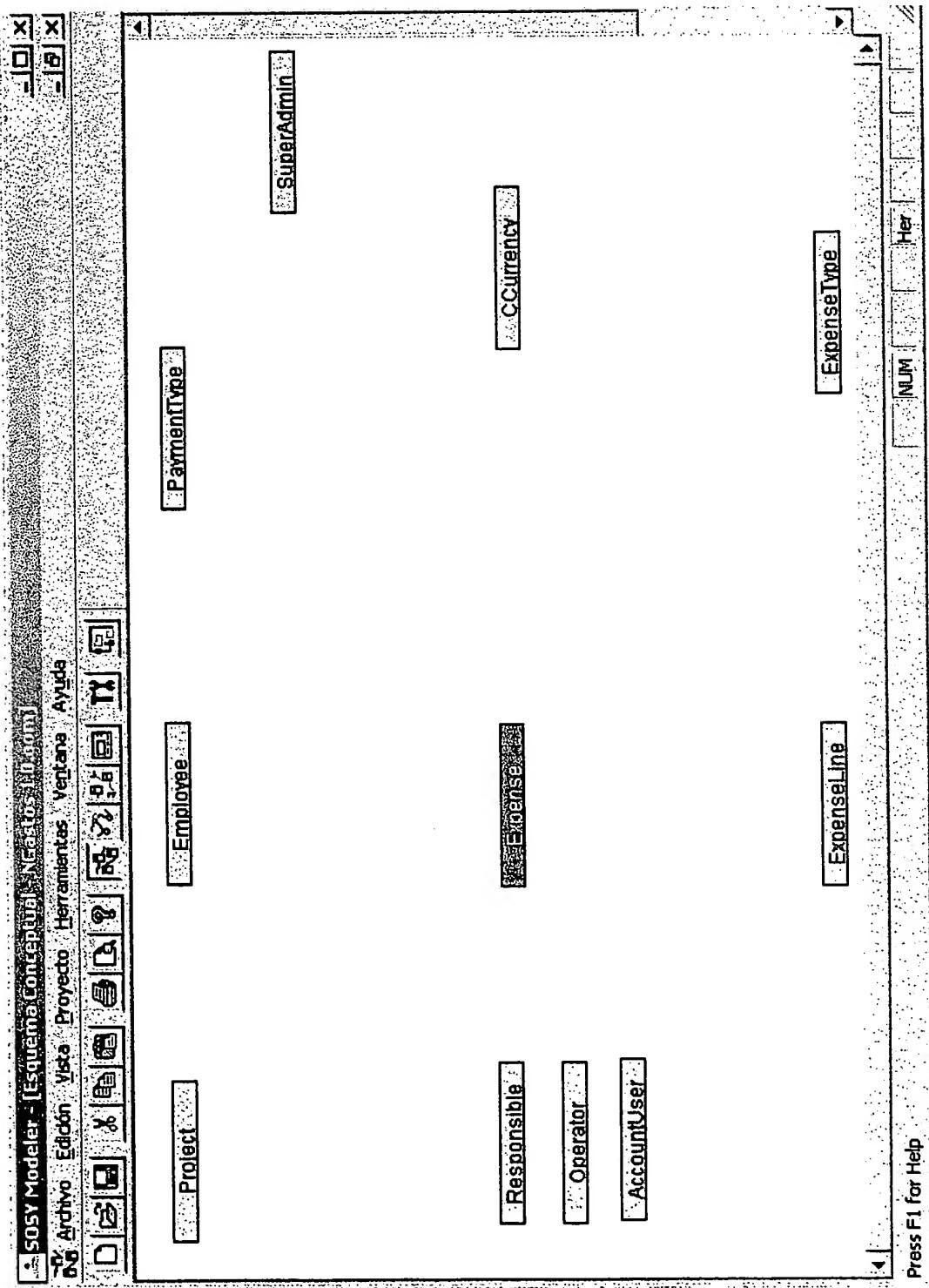
- Classes
- Relationships
- Attributes
- Services
- ...

Expressed in a non-ambiguous language.

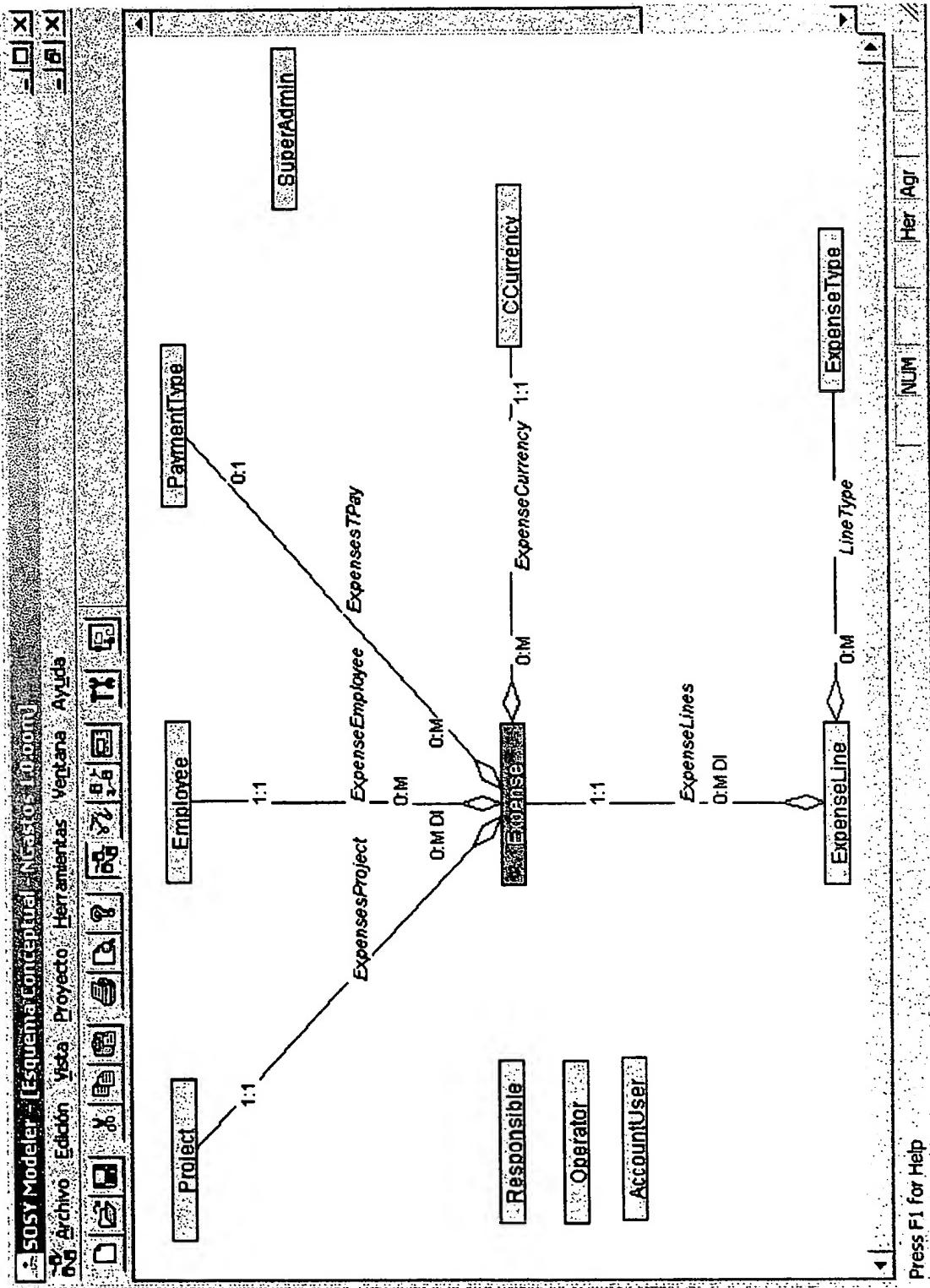
Phase 0. Requirement elicitation.

- Gathering the system requirements.
 - By meetings & interviews with customers, experts and final users.
 - By collecting reports, or documents expressing the system how-to and using tools.
 - Obtaining a coherent set of information as input to the next phase.

Phase 1. Classes identification.



Phase 2. Relationships between classes.



Phase 3. Filling classes' details.

Clase:						
<input type="button" value="Atributos"/> <input type="button" value="Servicios"/> <input type="button" value="Derivaciones"/> <input type="button" value="Restricciones"/> <input type="button" value="Agentes"/> <input type="button" value="Transacciones"/> <input type="button" value="Relaciones"/> <input type="button" value="Generidades"/>						
Atributos						
Nombre	Tipo Atributo	Tipo dato	Id	Tamaño	Valor defecto	Permitir crear
PresentDate	Constante	Date			today()	Sí
Status	Variable	Int			0	No
Cause	Variable	String			NULL	No
AuthoDate	Variable	Date			NULL	No
AuthoComments	Variable	String			NULL	Sí
PaymentDate	Variable	Date			NULL	Sí
PayComments	Variable	String			NULL	Sí
TotalExpenses	Derivado	Real			0	No
TotalExpensesCuit	Derivado	Real			0	Sí
Advances	Variable	Real			0	No
AdvancesCuit	Derivado	Real			0	Sí
Exchange	Variable	Real			0	No
Balance	Derivado	Real			0	Sí
BalanceCuit	Derivado	Real			0	No
Observaciones:						
<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>						

Phase 3. Filling classes' details.

Phase 3. Filling classes' details.

Phase 3. Filling classes' details.

The screenshot shows a software interface for managing expenses. At the top, there is a navigation bar with links: Clase, Atributos, Servicios, Direcciones, Restricciones, Agentes, Transacciones, Relaciones, and Generales. Below the navigation bar, there is a toolbar with buttons for 'DELETEALL', 'Formulas', and 'Transaction'. A message box displays the text: 'FOR ALL Lines DO Lines deleteLine(Lines) . delExpense(THIS)'. The main area is titled 'Expense' and contains several input fields and dropdown menus:

- Acción:** A dropdown menu with options: 'Aprobar', 'Rechazar', 'Parametros', and 'Inicializar'.
- Clase/Rot:** A dropdown menu with options: 'Servicio', 'Parametros', 'P_ThisExpense', and 'Inicializar'.
- Expense:** A dropdown menu with options: 'approve', 'reject', 'parameters', and 'initializar'.
- Agentes:** A dropdown menu with options: 'Servicio', 'Parametros', 'P_ThisExpense', and 'Inicializar'.
- Observaciones:** A text input field.
- Class:** A dropdown menu with options: 'Expense' and 'Cancelar'.
- Accept:** A button labeled 'Aceptar'.

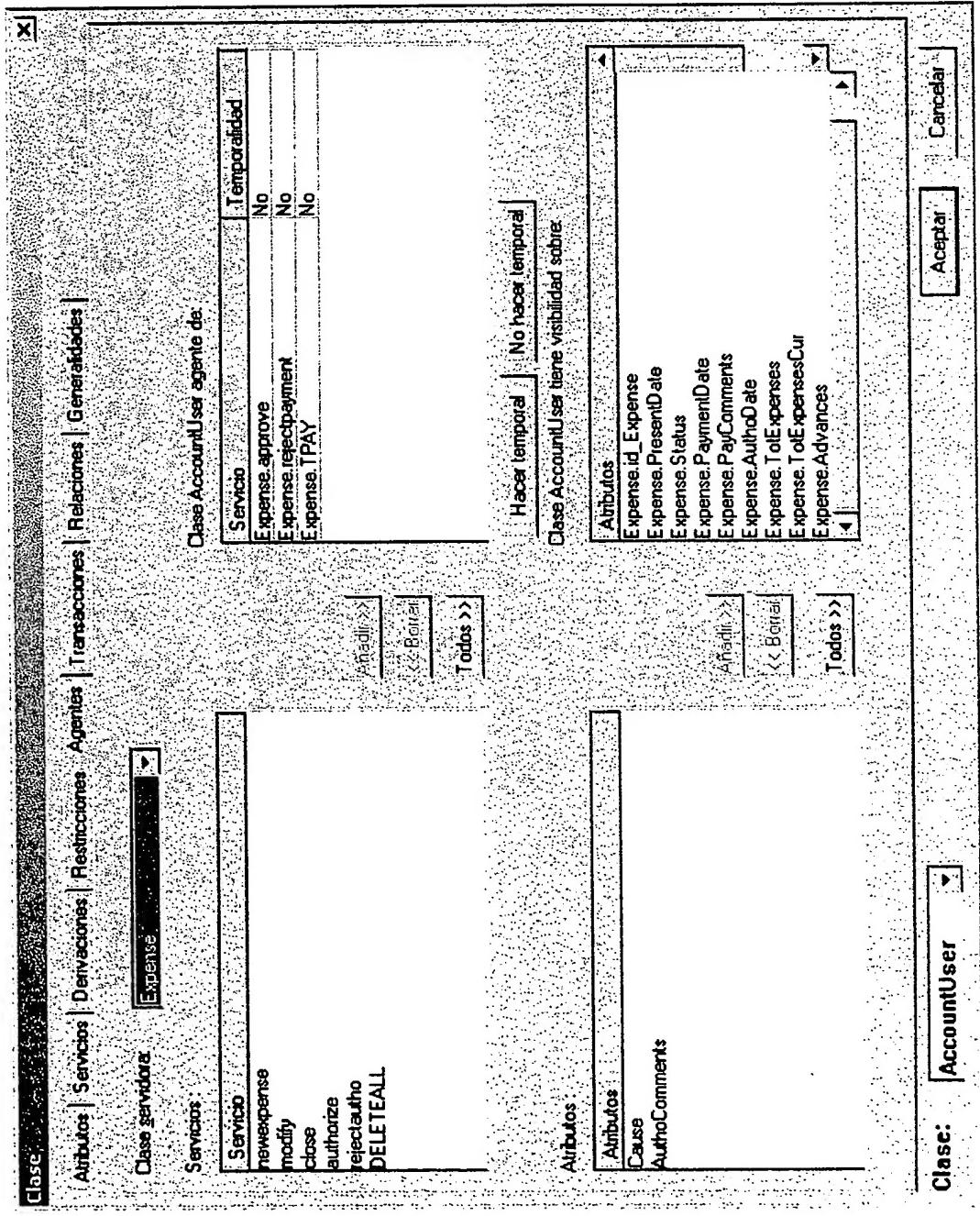
Phase 3. Filling classes' details.

Clase:	
Atributos	Servicios
Restricciones	Agentes
Transacciones	Relaciones
Generadas	Generadas
<hr/>	
Estáticas	Dinámicas
<hr/>	
Expense	Dinámicas
<hr/>	
Exchange > 0	Oper. Temporal
Formular	Condición1
Mensaje De Error:	Condición2
<hr/>	
Oper. Temporal	Oper. Temporal
Fórmula:	Formular
Mensaje De Error:	Mensaje De Error:
<hr/>	
Clase:	Expense
<hr/>	
Aceptar	Cancelar

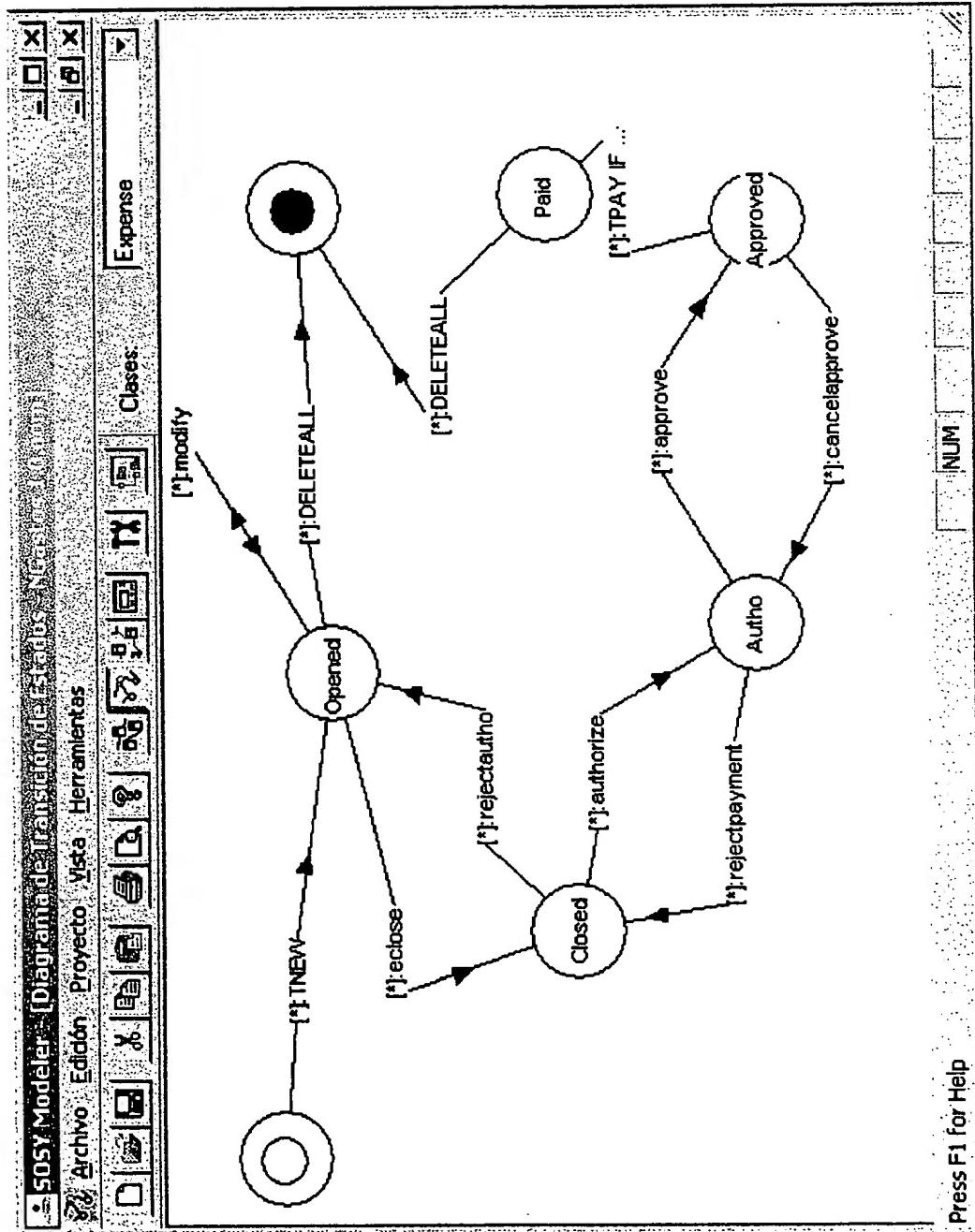
Phase 4. Express evaluations.

Modelo Funcional							
<input type="text" value="Expense"/> ▼	<input type="button" value="Autivo Cause"/> ▶						
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Evento</td> <td style="width: 30%;">Efecto</td> <td style="width: 40%;">Condición</td> </tr> <tr> <td><input checked="" type="radio"/> modificar</td> <td><input checked="" type="radio"/> p_Cause</td> <td><input checked="" type="radio"/> p_Condition</td> </tr> </table>		Evento	Efecto	Condición	<input checked="" type="radio"/> modificar	<input checked="" type="radio"/> p_Cause	<input checked="" type="radio"/> p_Condition
Evento	Efecto	Condición					
<input checked="" type="radio"/> modificar	<input checked="" type="radio"/> p_Cause	<input checked="" type="radio"/> p_Condition					
<input type="button" value="Aceptar"/> Cancelar							
<input type="button" value="Añadir"/> Modificar							
<input type="button" value="Borrar"/>							
[] Intent para el resto de atributos							
<input type="radio"/> Categoría							
<input type="radio"/> De Estado							
<input type="radio"/> De Situación							
<input type="radio"/> Cardinal							
<input type="radio"/> Evento							
<input type="radio"/> modify							
<input type="radio"/> Condición de evaluación							
<input type="radio"/> Efecto del evento							
<input type="radio"/> P_Cause							

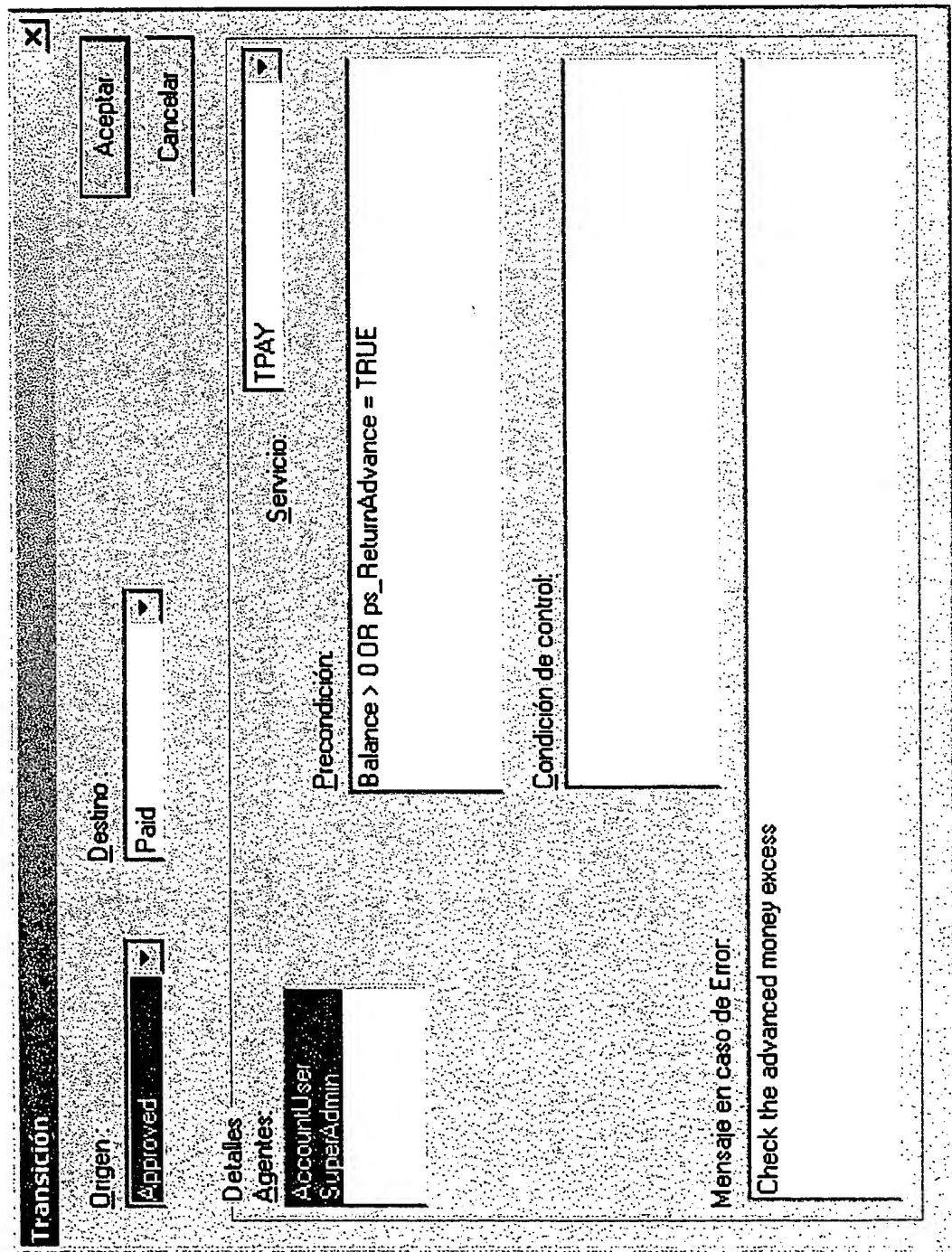
Phase 5. Agent relationships.



Phase 6. State Transition Diagram.



Phase 6. STD Preconditions



Phase 7. Presentation Model.

Conjunto de Visualización

Atributos a visualizar		Atributos	
Atributo	Tipo dato	Atributo	Tipo dato
Project.ProjectName	String	Cause	Date
Employee.EmpName	String	AuthComments	String
Employee.EmpSur...	String	PaymentDate	Date
Status	Int	PayComments	String
AuthoDate	Date	TotalExpenses	Real
PaymentDate	Date	TotalExpensesCur	Real
TotalExpenses	Real	Advances	Real
Balance	Real	AdvancesCur	Real
		Exchange	Real
		Balance	Real
		BalanceCur	Real

Nombre:

<< Eliminar >>

Clase: Expense

Phase 7. Presentation Model.

Filtro					
Fórmula:	<input type="text" value="vt_Project AND Employee = vt_Employee AND PresentDate >= vt_DateInIssue AND PresentDate <= vt_DateEndIssue AND AutoDate >= vt_DateInApp AND AutoDate <= vt_DateEndApp AND PaymentDate >= vt_DateInPay AND PaymentDate <= vt_DateEndPay AND"/>				
Alias:	<input type="text" value="Expense Reports"/>				
Borrar	<input type="button" value="<< Variable"/>				
Observar					
Variables					
Nombre	Alias	Tipo dato	Tipo estilo	Estilo	Nueva
vt_Project	Project	Project	Sel. Población		
vt_Employee	Employee	Employee	Sel. Población	Modificar	
vt_DateInIssue	InitialIssuing Date	Date			
vt_DateEndIssue	FinalIssuing Date	Date			
vt_DateInApp	Initial Approving D	Date			
Tipo		Estilo de introd.			
<input checked="" type="radio"/> Simple		<input type="text"/>			
<input type="radio"/> Objeto-valorado		<input type="text"/>			
Clase: Expense		<input type="text"/>			
		<input type="button" value="Aceptar"/>			
		<input type="button" value="Cancelar"/>			

CARE Technologies, S.A.

Conceptual Model Validation

Index

- Intro
- Overview
- Validation Degrees
 - Partial Validation
 - Total Validation

Index

- Validation Types
 - Elements of the Conceptual Model
 - Formulas of the Conceptual Model (Syntax)
- Validation Trees
 - Nodes
 - Leaves
- Example

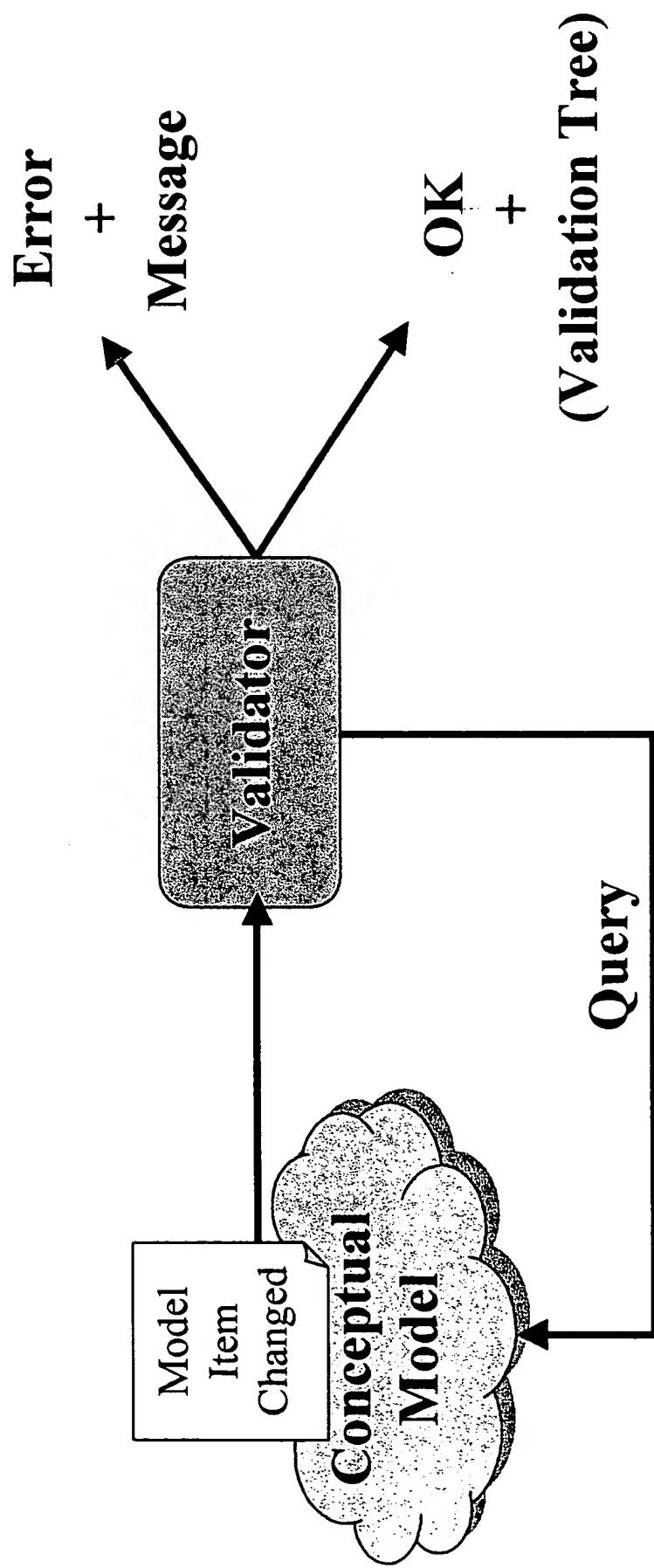
Intro

- Conceptual Model Validation is the process by which a conceptual model or a modification of it is proven to be valid:
 - Correct
 - Non Ambiguous
 - Non Contradictory
 - Complete
 - Every concept is fully specified
 - Validation process checks the representation of requirements in Formal Specification Language to be valid

Validation Degrees

- Partial Validation
 - That of a single element of the Conceptual Model.
 - Happens whenever an element is added, modified or deleted.

Partial Validation Overview

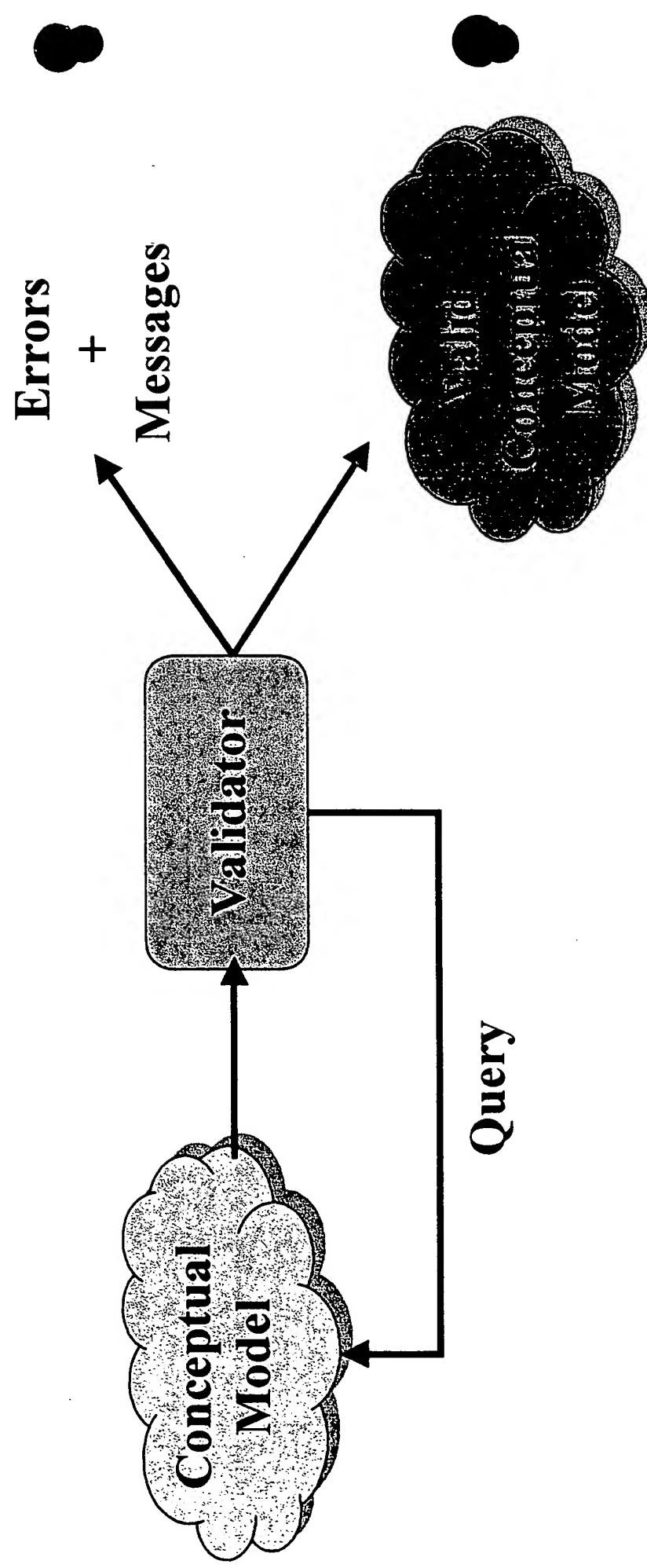


APPENDIX A

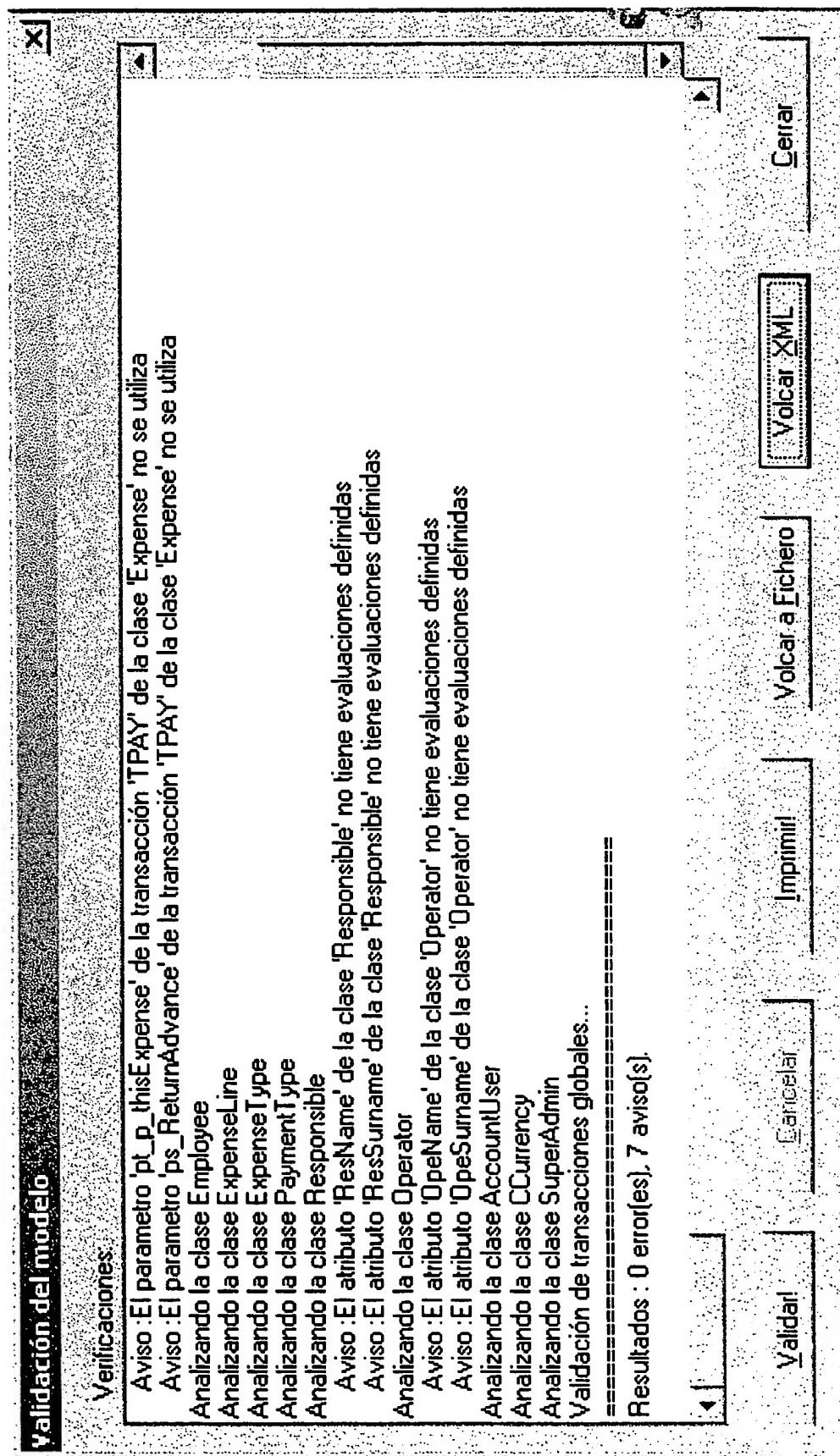
Validation Degrees

- Total Validation
 - That of the whole Conceptual Model.
 - Happens by request.
 - Must happen prior to any translation process.
 - Takes advantage of partial validations already performed.

Total Validation Overview



Total Validation Example



Validation Types

- Elements of the Conceptual Model
 - Ensure the properties of an element (except formulas) are correct and complete.
 - Conditions that must hold depend on the type of element and the property being validated.
 - Examples:
 - Class Name is unique in a Conceptual Model.
 - Attribute Name is unique in its Class (but not in a Conceptual Model)

Validation Types

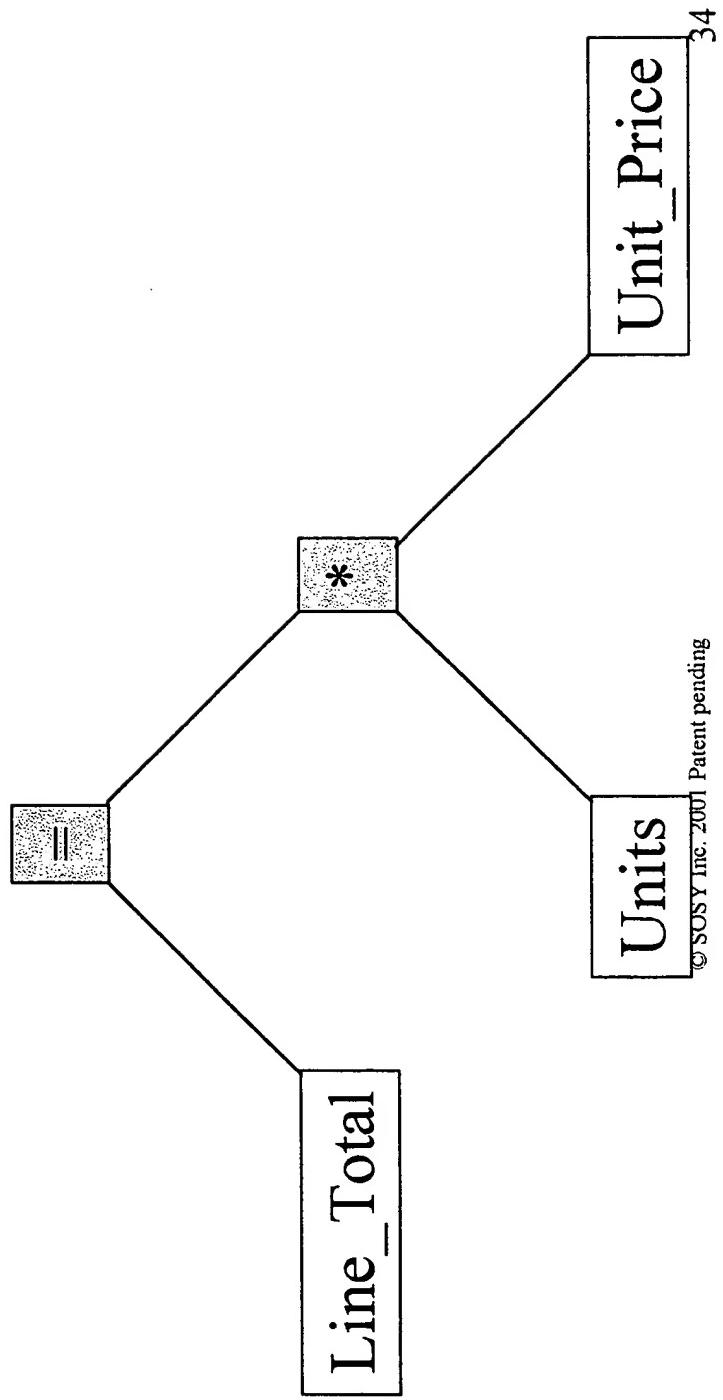
- Formulas of the Conceptual Model
 - Ensure the formulas of the Conceptual Model are correct and complete.
 - Syntactical and Semantical Validation according to an extended Formal Specification Language grammar.
- Input:
 - Formula expression
 - Formula Type (precondition, valuation, ...etc.)
 - Formula Context (class name, service name, ...etc.)
- Output:
 - Error Message (validation did not pass)
 - Validation Tree (validation passed)

Validation Trees

- Binary Tree representation of a correct formula.
- Tree consists of Nodes and Leaves.
- Nodes
 - Represent operators
 - Can have one or two “branches” (binary)
 - Branches can again be nodes or leaves
- Leaves
 - Represent operands
 - Have no branches

Example

- $\text{Line_Total} = \text{Units} * \text{Unit_Price}$



©SUSY Inc. 2001 Patent pending

CARE Technologies, S.A.

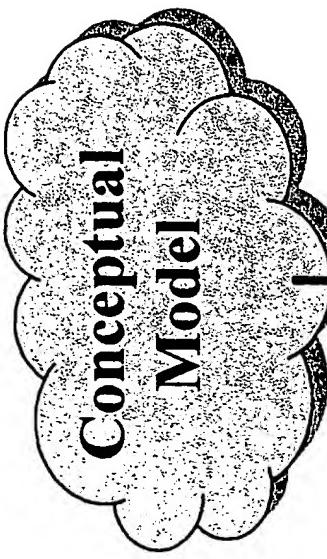
Documentation Translation

Index

- Intro
- Overview
- Output Detail
 - Document Types
 - Document Formats
- Translation
 - CM Subset of Interest
 - Translation Process
 - Remarks
- Example

Intro

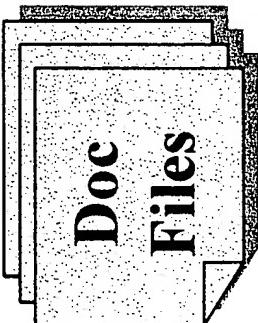
- Documentation Translation is the process to obtain, from a Conceptual Model, documentation on the system it represents.
- Documentation can have several degrees of detail and be focused on different aspects, thus obtaining different documentation formats from the same Conceptual Model.



Overview

Document Type

- Help
- Full
- General
- User Help Manual
- Project Report
- Test Report
- Multifile HTML
- Single File HTML
- ASCII Text
- LaTeX
- RTF
- ^{© SOSY Inc. 2004 Patent Pending} Compiled HTML



Output Detail

- Document Types
 - Help
 - Description of each Class, its Attributes, Services and Population Selection Filters.
 - Full
 - Full description of a Conceptual Model
 - Aimed at analysts.
 - General
 - Description of each Class Attributes, Identification Function, Services, Aggregation Relationships and Specialization Relationships.

Output Detail

- Document Types
 - User Help Manual
 - Both Help Manual and Contextual Help (F1 key).
 - Intended for Operation Manual.
 - Integration with User Interface applications.
 - Project Report
 - Description of each Class Attributes and Services.
 - Test Report
 - Description of each Class Services.
 - Intended for Testing purposes.

Output Detail

- Document Formats
 - Multifile HTML
 - One HTML page per concept.
 - Recommended for navigable help.
 - Single File HTML
 - One single HTML page.
 - Recommended for printing.
 - ASCII Text
 - Single, plain ASCII text file.

Output Detail

- Document Formats
 - LaTeX
 - Single, LaTeX text file.
 - RTF
 - Single, RTF text file.
 - Compiled HTML
 - Same as Multifile HTML plus header files to be used by HTML Help Workshop compiler.
 - Recommended for contextual help.
 - Searching and Indexing facilities usage from browsers.

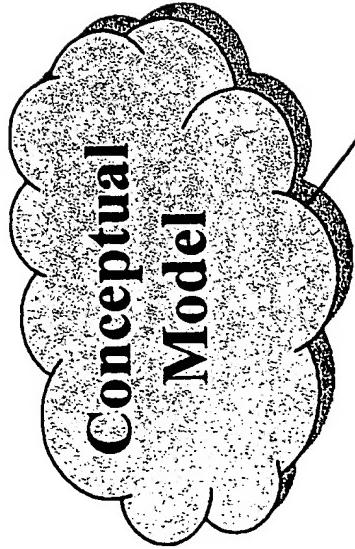
Translation

- Conceptual Model Subset of Interest
 - Subset of Interest depends on Document Type.
 - Usual elements:
 - Classes
 - Attributes
 - Relationships
 - Services & Arguments
 - Intensive use of analysis information.

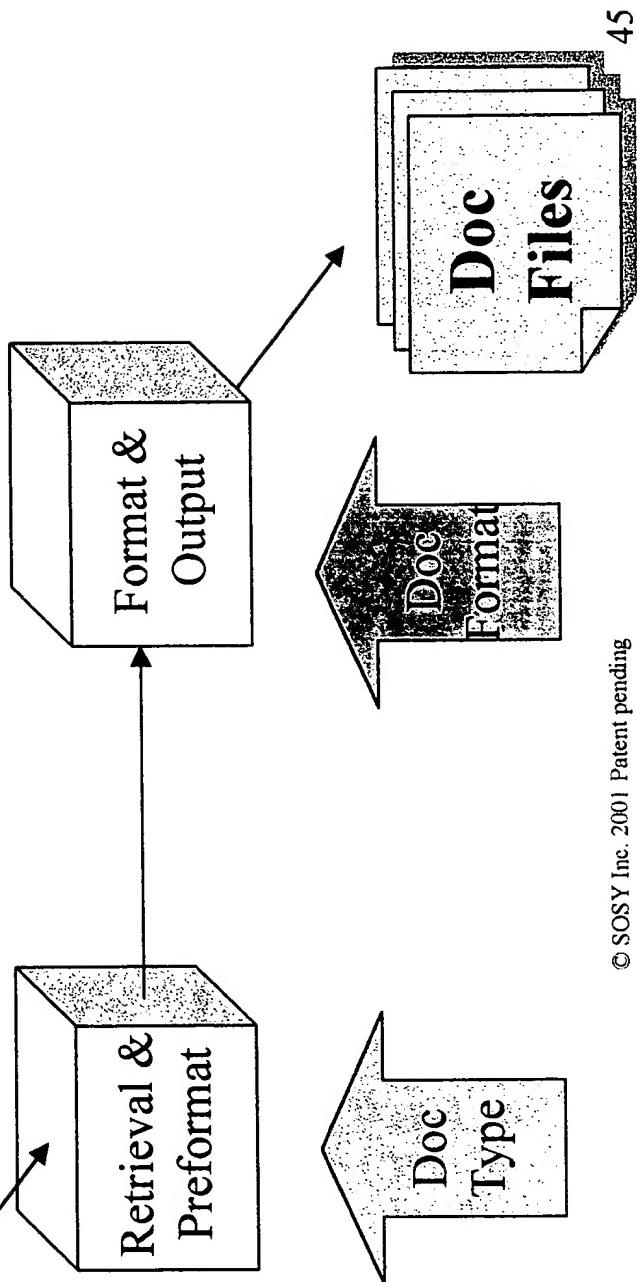
Translation

- Translation Process
 - Read information from Conceptual Model and format it for output.
 - Two phases:
 - Information retrieval and pre-formatting.
 - Depends on Document Type
 - Independent from Document Format
 - Information output.
 - Depends on Document Format.
 - Independent from Document Type.

Translation Phases



APPENDIX A



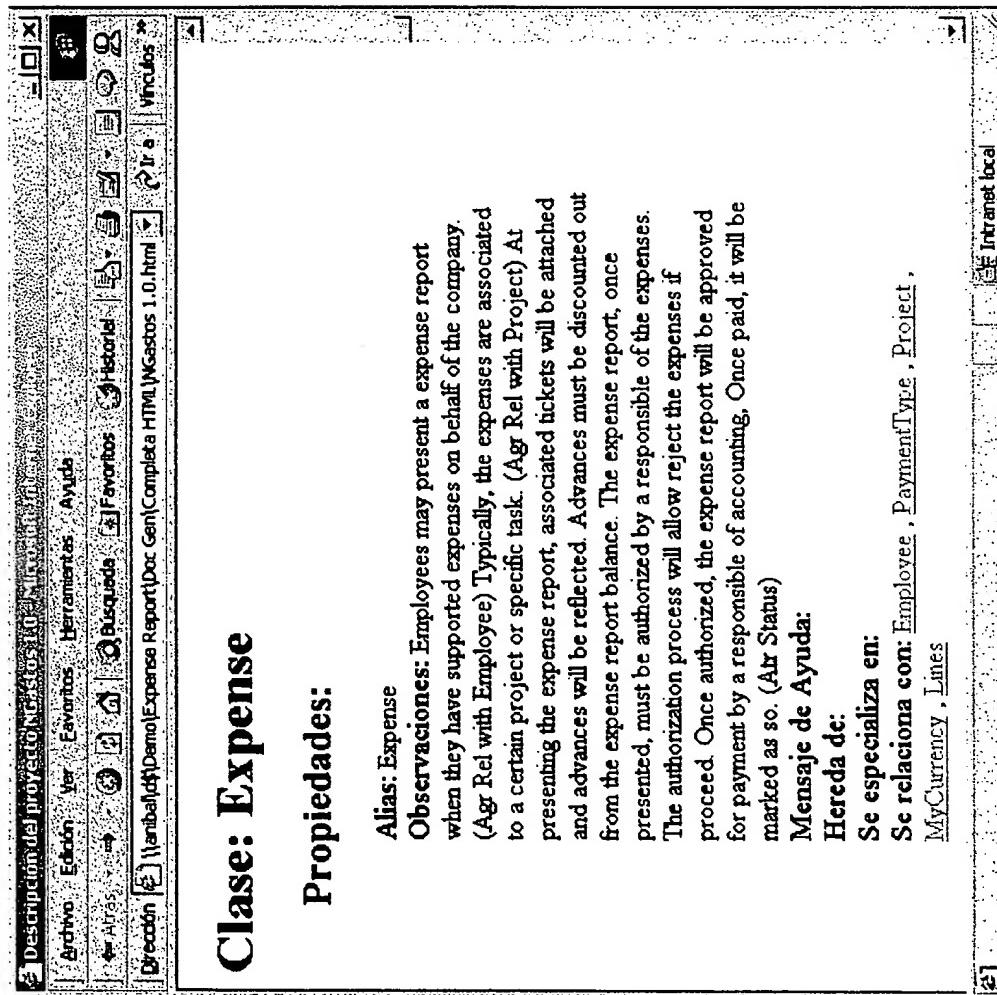
© SOSY Inc. 2001 Patent pending

Translation

- Remarks

- Conceptual Model needs not to be valid (in terms of completeness and correctness) but it is always non-ambiguous.
- The richer the analysis information, the richer the documentation.
- Easily extensible
 - New Document Types
 - New Document Formats

Example



Persistence Relational Database Translation

CARE Technologies, S.A.

Index

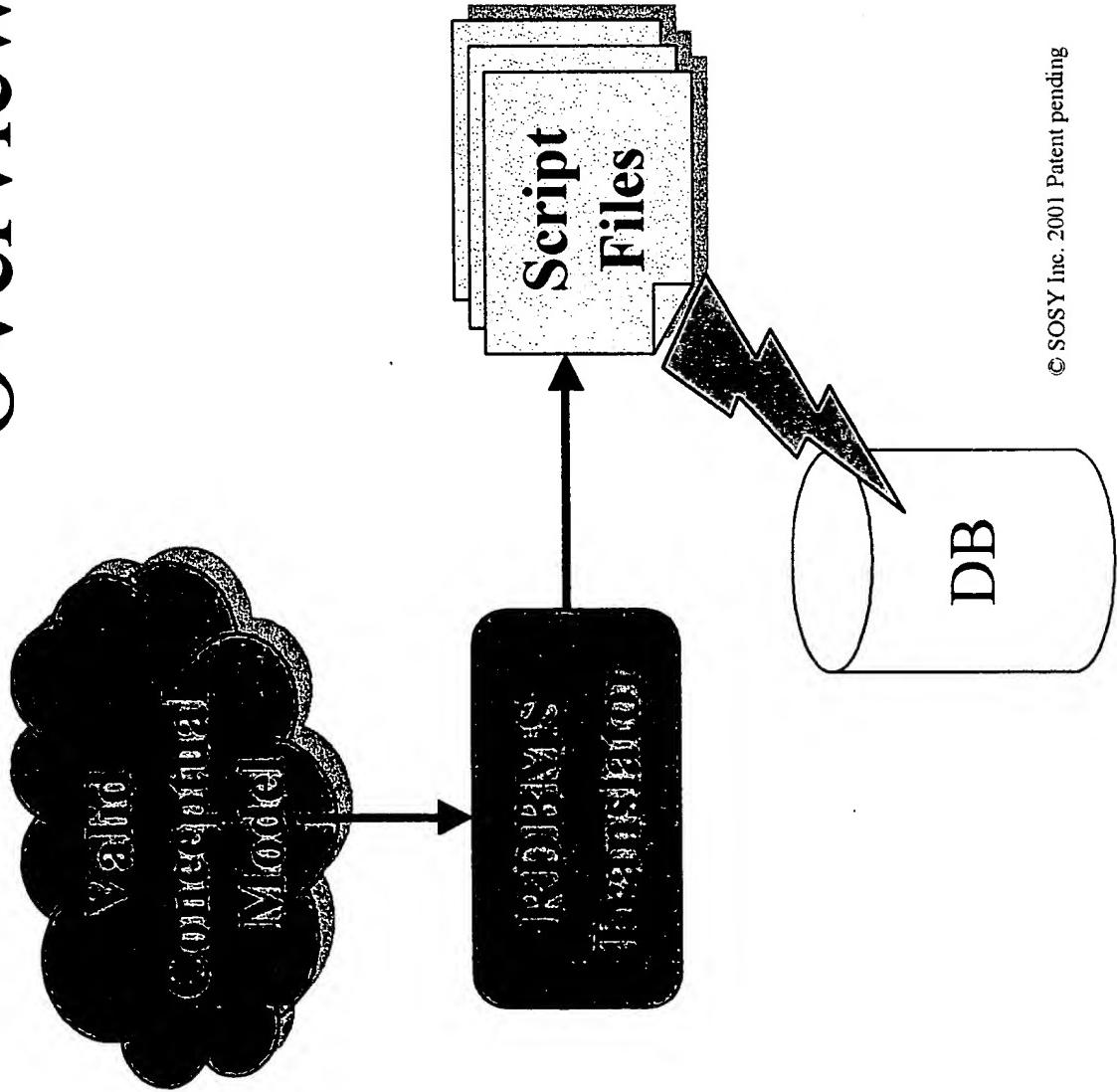
- Intro
- Overview
- Output Detail
- Translation
 - CM Subset of Interest
 - Translation Processes
- Example

Intro

- Persistence Relational Database Translation is the process of creating a Relational Database from a certain subset of information in the Object Model of a valid Conceptual Model.
- Output script files are used to create a relational database using structured query language (SQL).

Overview

- Creates
- Primary Keys
- Foreign Keys
- Indexes
- Drop Creates
- Drop Primary Keys
- Drop Foreign Keys
- Drop Indexes



© SOSY Inc. 2001 Patent pending

Output Detail

- Creates
 - Creation of Tables and Fields
- Primary Keys
 - Creation of Primary Keys as constraints on each table
- Foreign Keys
 - Creation of Foreign Keys as constraints on each table
- Indexes
 - Creation of Indexed on each table

Output Detail

- Drop Creates
 - Deletion of Tables
- Drop Primary Keys
 - Deletion of Primary Key Constraints
- Drop Foreign Keys
 - Deletion of Foreign Key Constraints
- Drop Indexes
 - Deletion of Indexes

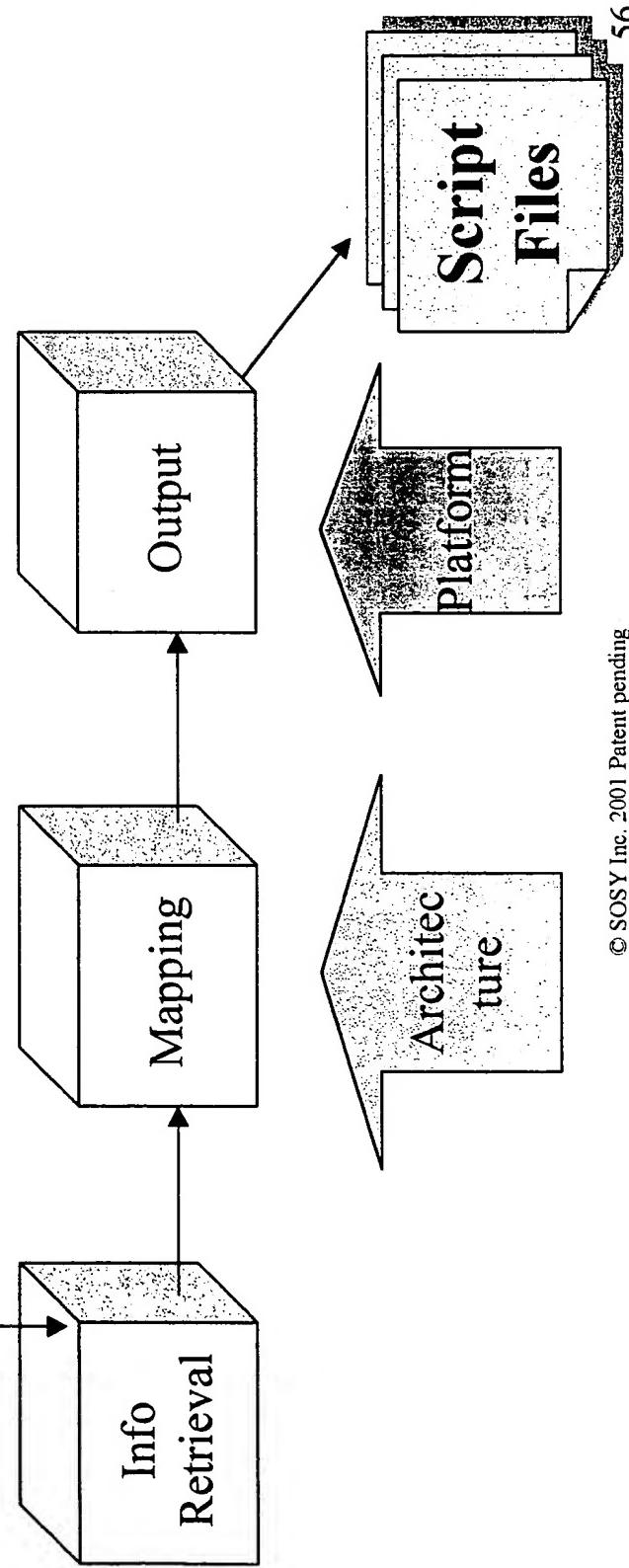
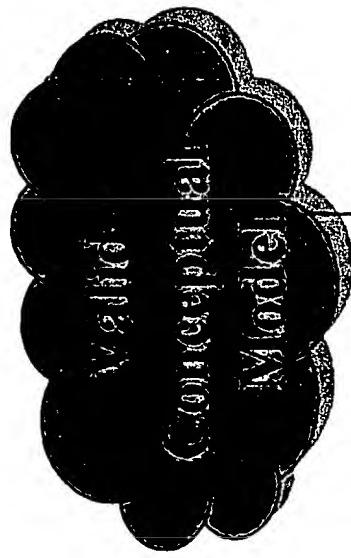
Translation

- Conceptual Model Subset of Interest
 - Object Model
 - Classes
 - Attributes
 - Identification Functions
 - Aggregation Relationships
 - Inheritance Relationships

Translation

- Three phases:
 - Information retrieval.
 - Independent from persistence architecture.
 - Fixed architecture mapping.
 - Depends on persistence architecture.
 - Information output.
 - Targeted for Standard ANSI SQL 92 RDBMS.
 - Script files depends on the platform's SQL syntax of RDBMS manufacturer.
 - May depend on platform specifications to make use of manufacturer extensions and tuning.

Translation Phases



Translation

- Translation Processes. Mapping:
 - Class → Table
 - Non-derived Attribute → Field
 - Identification Function → Primary Key
 - Univaluated Relationship → Foreign Key
 - Univaluated Relationship → Index
 - Multivaluated Relationship → Table
 - Inheritance Relationship → Foreign Key

Example

Create table script in SQL for Expense class

```
CREATE TABLE Expense (
    fk_Project_1 int NOT NULL ,
    id_Expense int NOT NULL ,
    fk_Employee_1 CHAR (10) NOT NULL ,
    fk_MyCurrency_1 CHAR (5) NOT NULL ,
    fk_PaymentType_1 CHAR (5) NULL ,
    PresentDate datetime NOT NULL ,
    Status int NOT NULL ,
    Cause VARCHAR (255) NOT NULL ,
    AuthoDate datetime NULL ,
    AuthoComments VARCHAR (255) NULL ,
    PaymentDate datetime NULL ,
    PayComments VARCHAR (255) NULL ,
    Advances DECIMAL (19, 6) NOT NULL ,
    Exchange DECIMAL (19, 6) NOT NULL );
```

Business Logic Translation

CARE Technologies, S.A.

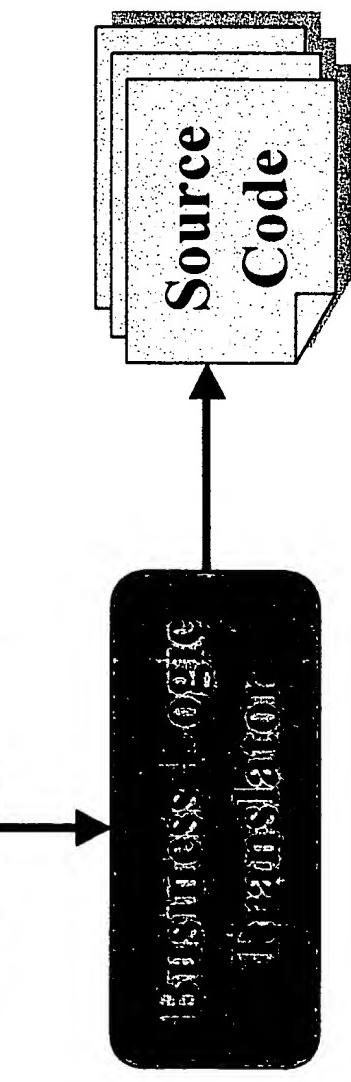
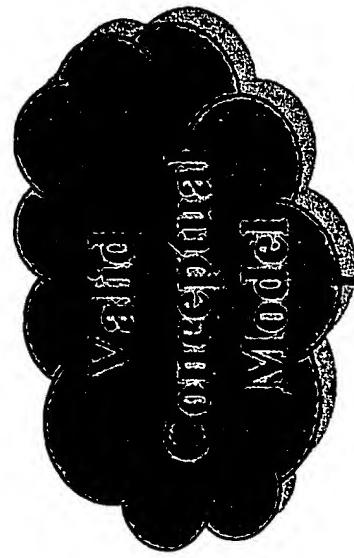
Index

- Intro
- Overview
- Output Detail
- Translation
 - CM Subset of Interest
 - Translation Processes
- Example

Intro

- Business Logic Translation is the process to obtain, following a precise Execution Model, the source code corresponding to the business logic from a valid Conceptual Model for a target Programming Language and Software Architecture.
- Execution Model is independent from Programming Language and Software Architecture.

Overview



Determines:

- *Target Programming Language*
- *Target Software Architecture*

© SOSY Inc. 2001 Patent pending

Output Detail

- Target Programming Language and Software Architecture determine:
 - Source code organization in files
 - Files internal organization
- Source Code's backbone: Execution Model.

Output Detail

- Traceability: Source code highly readable and maintainable thanks to:
 - Source code is always organized and structured in the same way.
 - Naming conventions applied.
 - Source code includes analysis information from the Conceptual Model as comments.

Output Detail

- Implementation of a precise Execution Model grants Functional Equivalence with Conceptual Model.
- Programming Interface to Clients for:
 - Actor Validation and Authentication.
 - Services Execution.
 - Queries Execution.
- Manages:
 - Concurrency.
 - Transactions.
 - Interoperable Objects Persistence.

Translation

- Conceptual Model Subset of Interest
 - Object Model
 - Static properties (Visibility & Persistence)
 - Attributes + Identification Functions
 - Derivations
 - Aggregation Relationships
 - Inheritance Relationships
 - Services (Execution Model)
 - Arguments
 - Preconditions
 - Transaction Formulas
 - Actors (Execution Model)
 - Integrity Constraints (Execution Model)

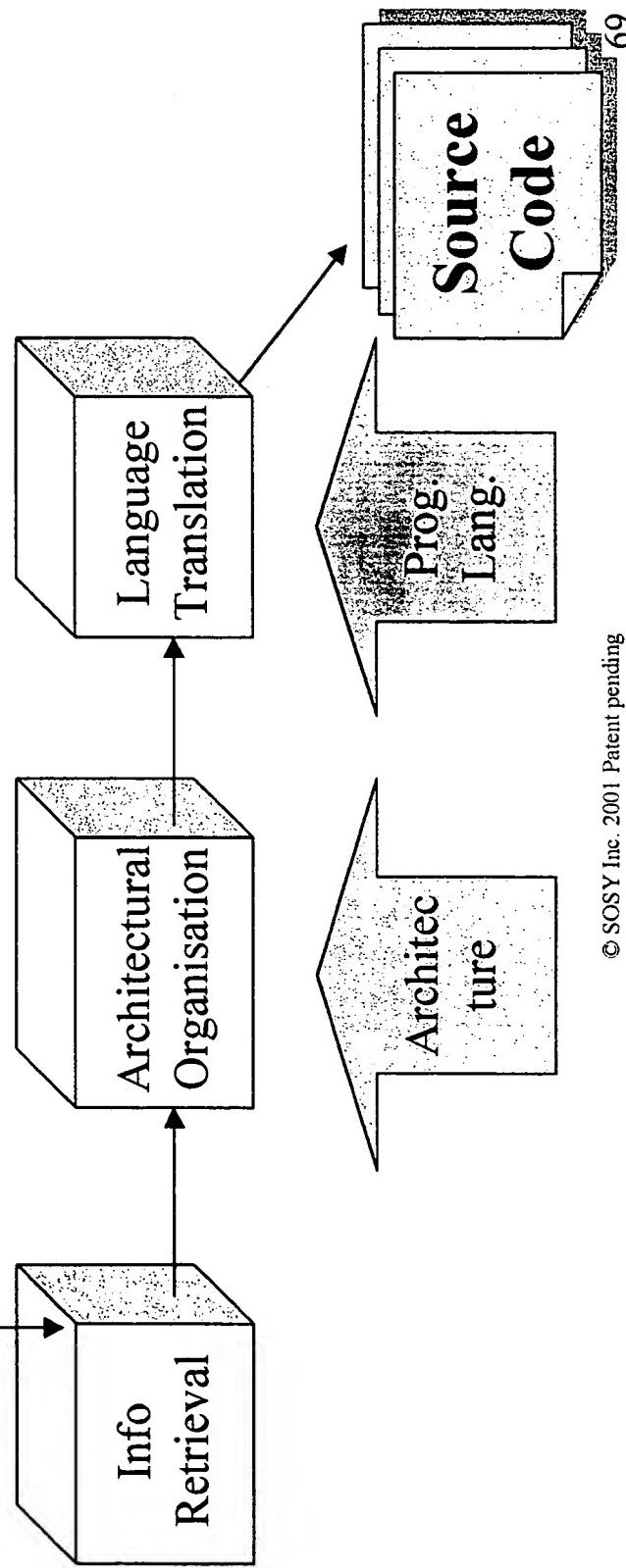
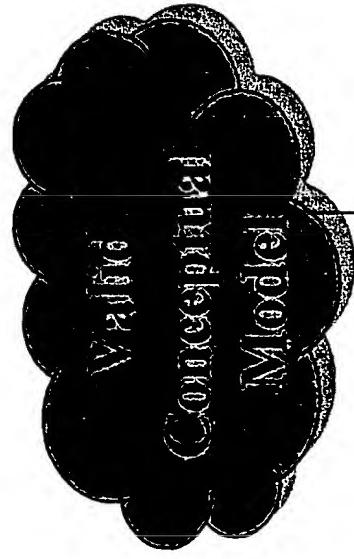
Translation

- Conceptual Model Subset of Interest.
 - Dynamic Model.
 - State Transition Diagram (Execution Model).
 - Controls Valid Lifes for an Object.
 - Object Interaction Diagram.
 - Triggers (Execution Model).
 - Global Transactions (Execution Model).
 - Functional Model (Execution Model).
 - Object state change upon occurrence of an event.

Translation

- Translation phases:
 - Information retrieval
 - Independent from target Software Architecture and Programming Language
 - Architectural organisation
 - Depends on target Software Architecture
 - Independent from target Programming Language
 - Determines files organisation and files internal structure
 - Language translation
 - Depends on target Programming Language
 - Influenced by Software Architecture
 - Takes advantage of Programming Language capabilities

Translation Phases



Translation

- Translation Processes
 - Classes
 - Static properties translation
 - Services translation
 - Queries translation
 - Global Interactions
 - Services translation
 - Global Functions
 - Functions Interface translation
 - Body is left blank

Example

- Evaluation:
 - Service Authorize modifies attributes Status, AuthoDate and AuthoComments
 - Formal Specification Language expression for evaluation Valuation
 - [authorize ()] Status=2 and AuthoDate=today() and AuthoComments="";
 - Visual Basic Produced

```
Private Function MV_Eval_Expense_authorize() As String  
Expense_Status = 2  
Expense_AuthoDate = today()  
Expense_AuthoComments = ""  
MV_Eval_Expense_authorize = ""  
End Function
```

© SOSY Inc. 2001 Patent pending

“ ସାମାଜିକ ଯୋଗାନ୍ତର ପାଇଁ ଏହାର ଉଦ୍ଦେଶ୍ୟ ହାତରେ ଥିଲା ।

User Interface Translation

CARE Technologies, S.A.